Introduction to CIAO

Chandra Interactive Analysis of Observations

CIAO: Chandra's data analysis system
Fruscione et al. 2006, SPIE Proc. 6270, 62701V,
D.R. Silvia & R.E. Doxsey, eds.

What is CIAO?

First a linguistic note….

CIAO ….. from "s'sciavo", "I am your servant" in Venetian language

And CIAO has been at the service of X-ray astronomers
for more than 25 years!

CIAO 1.0 was released in October 1999

CIAO 4.17 was released in December 2024

If you're new to CIAO or unsure about how something works, the best place to start is always the help system (or ahelp). Let's try it now.

To learn what CIAO is, you can access the help file directly from the command line by running:

```
ahelp ciao
```

or you can read the same information online at:
cxc.harvard.edu/ciao/ahelp/ciao.html

AHELP for CIAO 4.17 **ciao** Context: concept

Synopsis

Chandra Interactive Analysis of Observations

Description

The remarkable science capabilities of the Chandra X-ray Observatory demanded new, flexible, multi-dimensional, software to analyze the data it returned. The result is CIAO - the Chandra Interactive Analysis of Observations - a system that has proven itself useful for the analysis of data from other, non-X-ray missions, because of the mission independence that is the basis of the CIAO design. [...]

AHELP CIAO - SUMMARY

Purpose: CIAO (Chandra Interactive Analysis of Observations) is a software system designed for analyzing data from the Chandra X-ray Observatory.

Flexibility: It was developed to handle the new, flexible, multi-dimensional (initially 4D: 2 spatial, time, energy) data returned by Chandra.

Mission Independence: A key design feature of CIAO is its mission independence, making it useful for analyzing data from non-X-ray missions as well.

N-Dimensional Data Handling: CIAO tools are built to handle N-dimensional data without concern for the specific axes being analyzed.

File Format Compatibility: CIAO tools can read and write various formats, including FITS tables (event files), ASCII formats, and FITS images, allowing users to integrate pre-existing tools.

AHELP CIAO - SUMMARY (continue)

Filtering and Binning: The system allows for flexible filtering and binning of data (e.g., Chandra's 4D event data) into manageable sizes and arrays using a command-line 'regions' syntax. This is facilitated by the Data Model ("ahelp dm").

Data Subspace Tracking: CIAO keeps track of how data has been filtered and binned in a 'data subspace', which is automatically managed by the tools and allows users to review previous processing steps ("ahelp subspace").

Tool Interconnection: CIAO tools are designed for close interconnection, allowing output from one tool (e.g., source detection) to be used as input for another (e.g., `dmextract` for creating spectra, then fitting in Sherpa). This supports packages like `dax` for analysis directly from DS9.

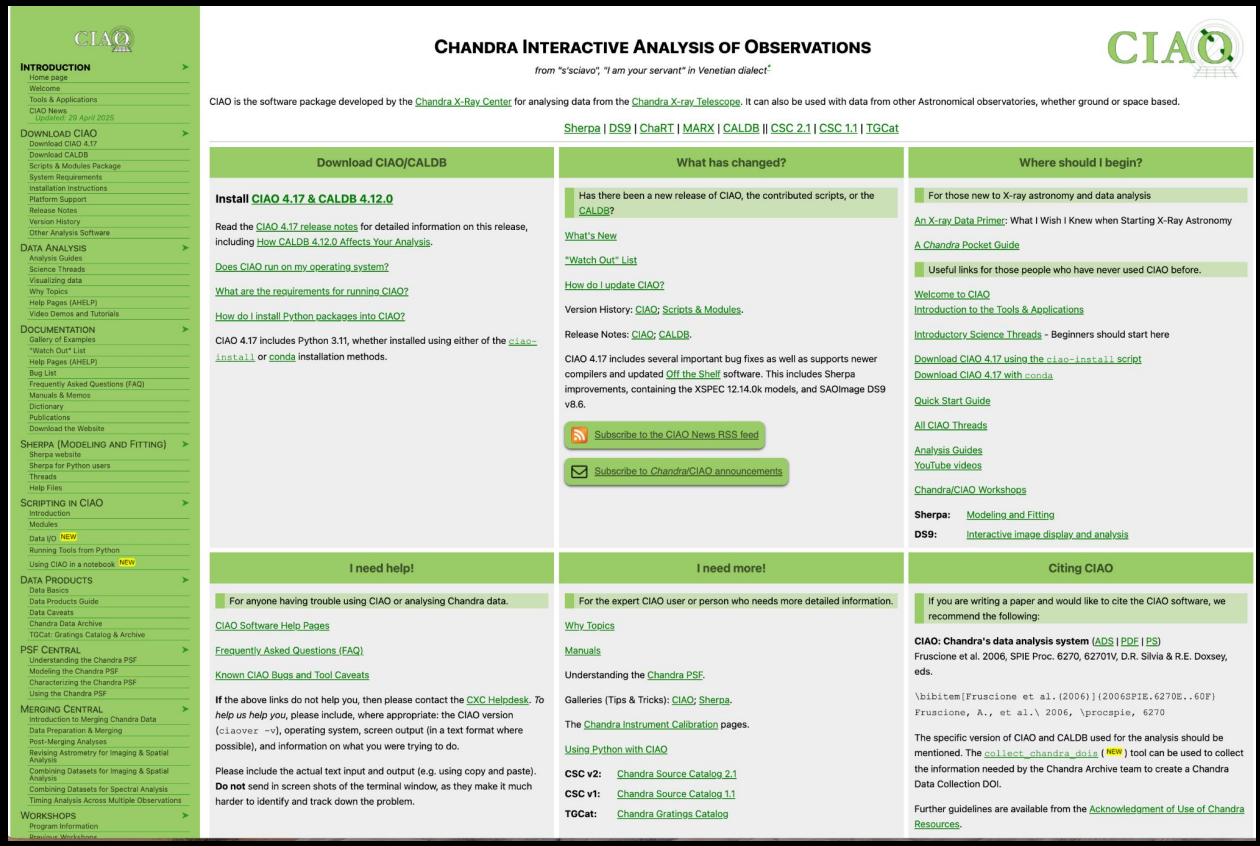
Modeling and Fitting with Sherpa: Sherpa is the central modeling and fitting tool within CIAO, capable of forward fitting models to N-dimensional data.

Resource for More Information: More details are available on the CIAO website and through CIAO's command-line help system, `ahelp`.

- A collection of Unix command line tools and Python applications
- Shares code with standard Chandra processing pipeline
- Allows Chandra instrument specific data reduction (eg ARF and RMF)
- Tailored to specialized X-ray astronomy data analysis, but not specific to Chandra (as been used with XMM, NuStar, etc.)
- Coded with attention to standards and interoperability so that generic tools can be (and are) used for other X-ray data and even optical and radio data (e.g. multiwavelength analysis)
- Easy for beginners, yet powerful for advanced users
- Linux and Mac, annual releases
- Installed 1500+ times per year (single users to large institutions)

What is CIAO in practice?

cxc.harvard.edu/ciao



General Concepts

1. File format
2. Parameter Files
3. Filtering and Binning (the Data Model)
4. Regions
5. Subspace
6. Good Times Intervals (GTI)
7. Scripting language (Python)

1. File Format

- Chandra data is stored in the **binary FITS format** (more on that later!), but many CIAO tools can also work with simple ASCII text files.
- When CIAO processes data, it **records processing state and metadata** directly in the file, using keywords and subspace information in the header so that history and information is preserved.
- A single Chandra FITS file **can include multiple datasets**, such as:
  - Event data
  - Good Time Intervals (GTIs)
  - Weight maps
  - Region definitions
- These datasets are stored in separate blocks within the file, each containing either image or table data.
- You can explore the contents of a file using:
  - `dmlist` — a command-line tool
  - `prism` — a graphical interface, which can be launched from DS9

The FITS format

Overview from https://heasarc.gsfc.nasa.gov/docs/heasarc/fits_overview.html

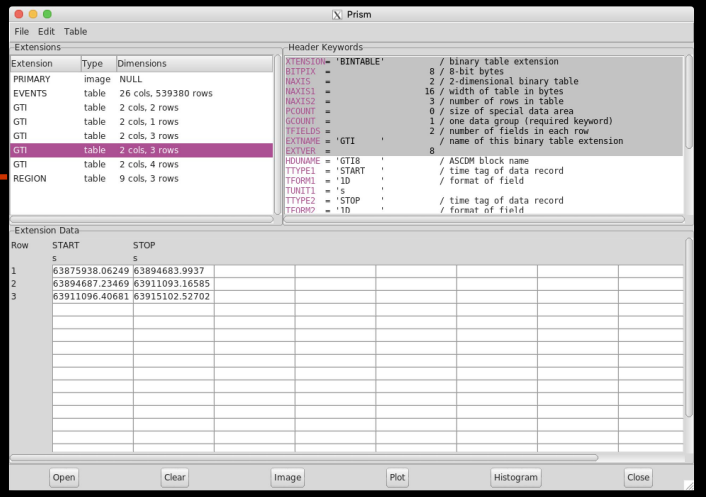
- FITS stands for *Flexible Image Transport System*.
- It is the **standard file format** used in astronomy to **store and share data**.
- A FITS file contains one or more "**data blocks**", each starting with a "**header**" followed by the "**data**".
  - The header is made of readable text and describes the structure and contents of the data.
  - The data can include images, tables, or other arrays (1D, 2D, or more).
- FITS can store different types of information in extensions, such as:
  - Images (e.g., sky maps or detector frames)
  - Tables (e.g., event lists or catalogs)
- FITS is **platform-independent** and designed to be long-term and self-describing, so files can be understood decades later.
- It is **supported by many tools and programming libraries** used in astronomy (e.g., CIAO, Astropy, fv).
- FITS is **maintained by an international standards group** to ensure consistency across the field.

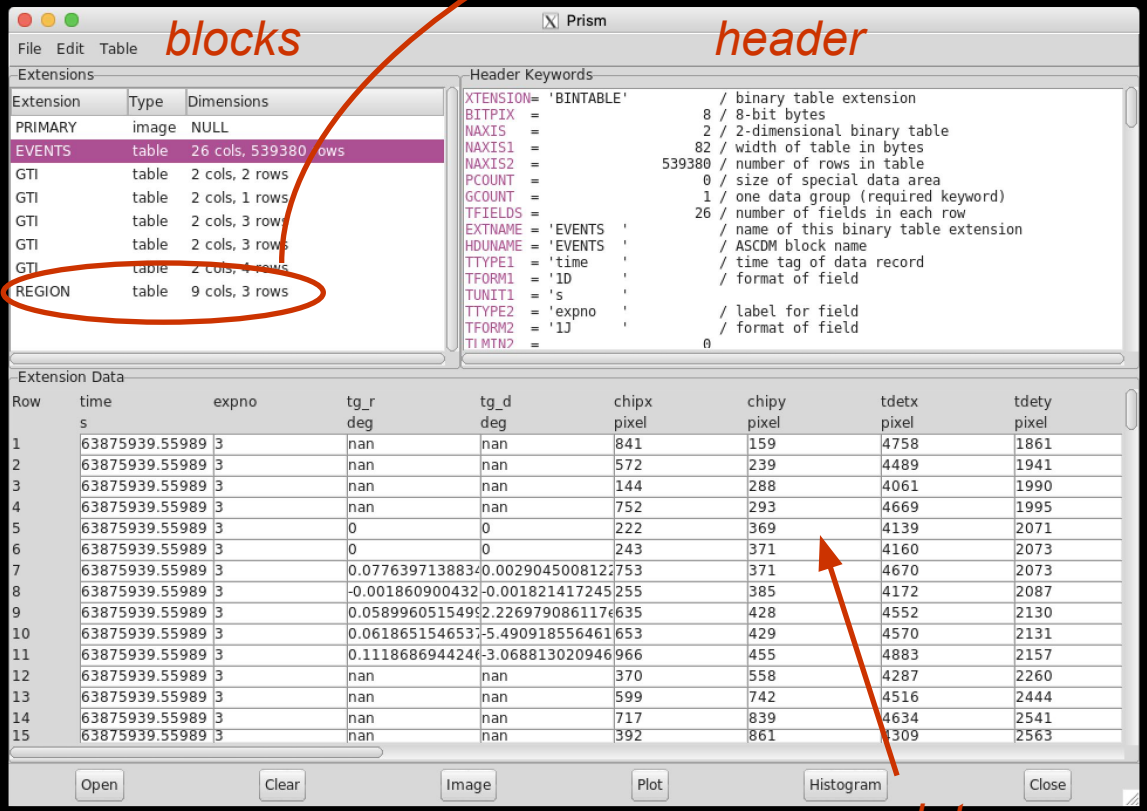
PRISM (in DS9) : examines FITS files, performs quick plots

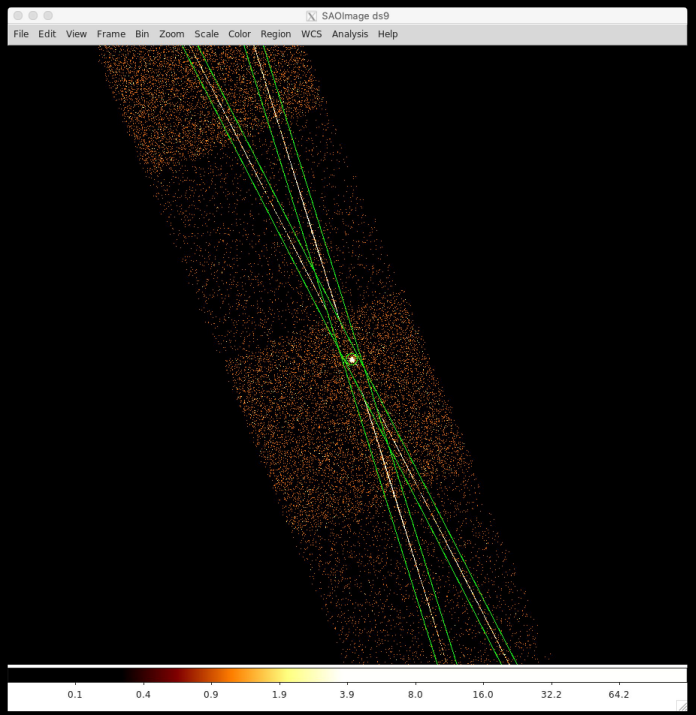
ciao> ds9 -prism acisf00459N004_evt2.fits

or

ds9 –> File → Prism







2. Parameter Files

From: http://cxc.harvard.edu/ciao/ahelp/parameter.html

- CIAO tools use ASCII parameter files (e.g., dmcopy.par, specextract.par) to **store and retrieve input settings** for data processing.
- Users can **manage parameters** using command-line routines like `plist` (view), `pset` (set), and `punlearn` (reset), or via the Python interface (`paramio` module).
- Each parameter has a **mode** (e.g., query, automatic, hidden) that determines how it is accessed or prompted during tool execution.
- Some parameters **accept multiple values (stacks),** as indicated in their help files (see `ahelp stack` for more on stack syntax).

Parameter file example: `dmcopy.par`

```
ciao% plist dmcopy

Parameters for /Users/antonella/cxcds_param4/dmcopy.par

        infile = acisf00459N004_evt2.fits.gz[REGION]  Input dataset/block specification
       outfile = grating_reg.fits                     Output dataset name
      (kernel = default)                              Output file format type
      (option = )                                     Option - force output type
     (verbose = 0)                                    Debug Level
     (clobber = no)                                   Clobber existing file
        (mode = ql)


ciao% dmcopy
Input dataset/block specification (acisf00459N004_evt2.fits.gz[REGION]):
Output dataset name (grating_reg.fits):
Clobber set to no, and output file grating_reg.fits exists.

ciao% pset dmcopy outfile=grating_reg2.fits
ciao% dmcopy

Input dataset/block specification (acisf00459N004_evt2.fits.gz[REGION]):
Output dataset name (grating_reg2.fits):
```

Note the required (infile, opt) and **optional** parameters (the ones in parenthesis)

3. Filtering and Binning (the “Data Model”)

http://cxc.harvard.edu/ciao/ahelp/dm.html
http://cxc.harvard.edu/ciao/ahelp/dmfiltering.html
http://cxc.harvard.edu/ciao/ahelp/dmbinning.html
http://cxc.harvard.edu/ciao/threads/filter/

- **Filtering and binning** are essential steps in X-ray data analysis.
  - Filtering is used to **remove unwanted events**—such as periods of high background, poor aspect solution, or irrelevant sources—from a dataset. This step helps ensure that the analysis focuses only on scientifically useful data.
  - Binning **transforms event lists into structured data** like histograms, 2D images, or 3D cubes. This is key for visualizing spatial, spectral, or temporal properties of the observation.
- These operations are powered by the **CIAO Data Model (DM)**—a versatile interface that allows users to work with both FITS and ASCII files using a consistent syntax. The Data Model hides the differences between file types, so a user can filter and bin data the same way no matter what kind of file they have.
- The Data Model is what makes CIAO tools powerful, portable, and format-independent.

BENEFITS OF THE DATA MODEL

- **Virtual File Access:**
  Any CIAO tool that takes a filename can also accept a *virtual file* string. This lets the tool operate on a filtered view of the data—without creating a new file on disk.

- **Optional File Creation:**
  The same syntax can also be used to create a filtered output file, if desired.

- **Flexible Binning and Filtering:**
  All columns in an event file are treated equally—so you can filter or bin on spatial coordinates, time, energy, or any other column.
  This enables the creation of multi-dimensional images, like space vs. energy or time vs. energy.

Data Model Syntax (ahelp dmsyntax)

All CIAO tools use the DM library and therefore accept "**virtual files**" as input

All CIAO tools use a common syntax to describe "**virtual files**":

filename[block][filter][columns/binning][options]

block – the data section to use (e.g. [EVENTS])

filter – the condition(s) applied to rows (e.g. [energy<7000])

columns/binning – which columns to keep, or how to bin data (e.g. [bin x=::4])

options – advanced settings for DM tools (rarely needed)

The **order of qualifiers generally matters**, but not all are required.

Let's see some examples!

Virtual File Examples (CIAO Data Model Syntax)

These examples show how to use CIAO's virtual file syntax to filter, select, and bin data from event files without creating intermediate files on disk.

Example 1: Select first 3 columns by number after time filtering

acisf01843N001_evt2.fits[EVENTS][time=84245787:84247000][cols #1,#2,#3]

Block: [EVENTS]
Filter: [time=84245787:84247000]
Columns: [cols #1,#2,#3]

Example 2: Select columns by name after filtering on grade

acisf01843N001_evt2.fits[EVENTS][grade=0,2,3][cols time,ccd_id,node_id]

Block: [EVENTS]
Filter: [grade=0,2,3]
Columns: [cols time,ccd_id,node_id]

Example 3: Create a PI spectrum for a specific region

acisf01843N001_evt2.fits[EVENTS][sky=region(mysrc.reg)][bin pi=1:1024:1]

Block: [EVENTS]
Filter: [sky=region(mysrc.reg)] *(the region)*
Binning: [bin pi=1:1024:1] *(the binning specification)*

Typical input to **dmextract**

Example 4: Create an image by filtering in energy and binning in (x,y) coordinates

acisf01843N001_evt2.fits[EVENTS][energy<7000][bin x=320:480:4,y=320:480:4]

Block: [EVENTS]
Filter: [energy<7000] *(the energy filter)*
Binning: [bin x=320:480:4, y=320:480:4] *(the binning specification)*

Typical input to **dmcopy**

Example: how to create a 2D image in sky coordinates from an ACIS event file using the dmcopy tool.

```
dmcopy "acisf06934N002_evt2.fits[bin x=3500:4500:2,y=3500:4500:2]"
6934_sky_binsize.fits
```

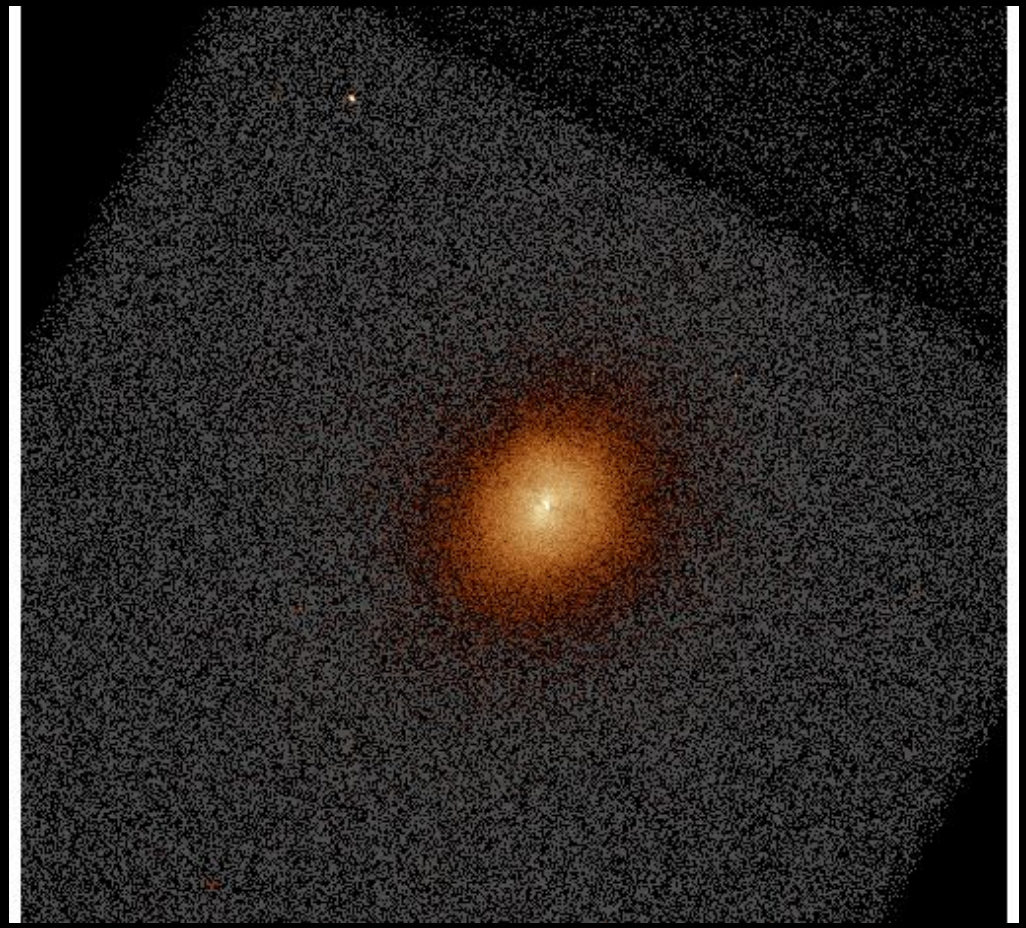
Input file: acisf06934N002_evt2.fits — an ACIS level 2 event file

Virtual file syntax:
[bin x=3500:4500:2, y=3500:4500:2]

Binning:
X-axis: from pixel 3500 to 4500 in steps of 2
Y-axis: from pixel 3500 to 4500 in steps of 2

This creates a 2D image in sky coordinates with reduced resolution (2-pixel bins)

Output file: 6934_sky_binsize.fits — the new binned image



Example: how to use spatial filtering with a region file in CIAO's dmcopy command.

```
dmcopy "ngc1404.img[sky=region(ngc1404_sample.reg)]" ngc1404_regfile.img
```

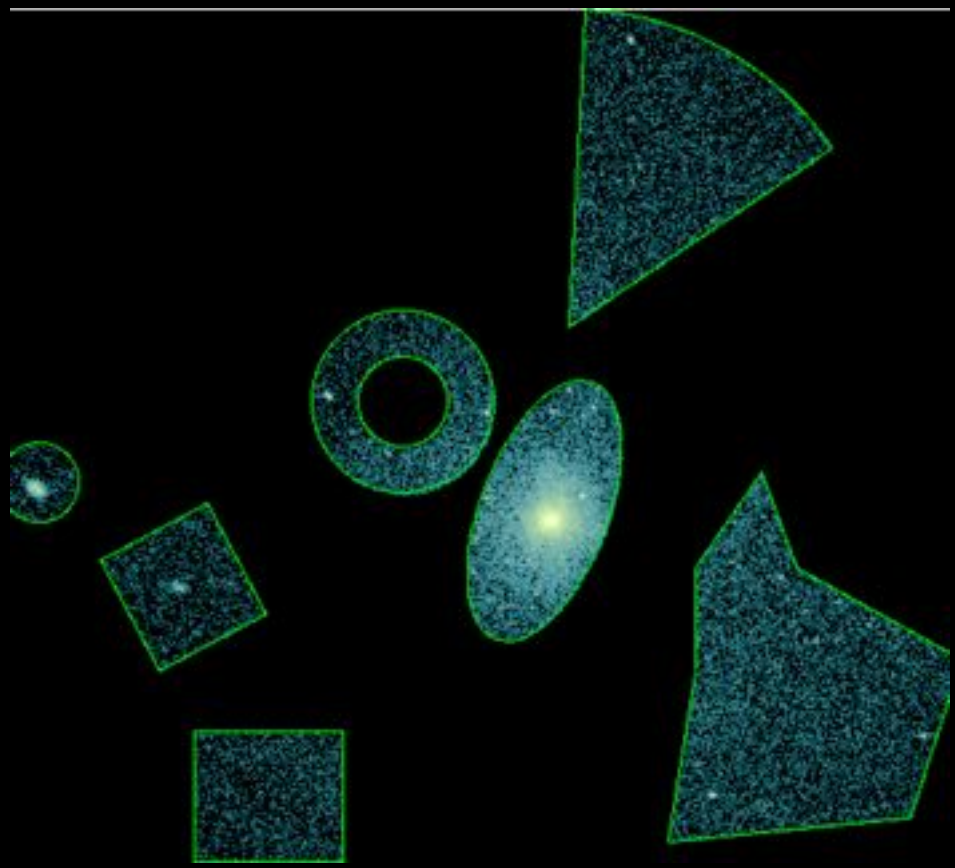
Input file: ngc1404.img — an image file

Filter: [sky=region(ngc1404_sample.reg)]

Applies a **spatial filter** using the region defined in *ngc1404_sample.reg*

The region file can define shapes like circles, boxes, ellipses, etc., in sky coordinates

Output file: ngc1404_regfile.img — the filtered image, containing only the pixels within the region



Example: Creating a 3D Image Cube

```
dmcopy "06540_evt.fits[(chipx,chipy)=box(8003.5,8137.5,512,512,0)] [bin
chipx=::2,chipy=::2,time=::#50]" outfile=cube.fits
```

This example creates a **3D image (or cube)** of a point source using **chip coordinates** from an HRC event file.

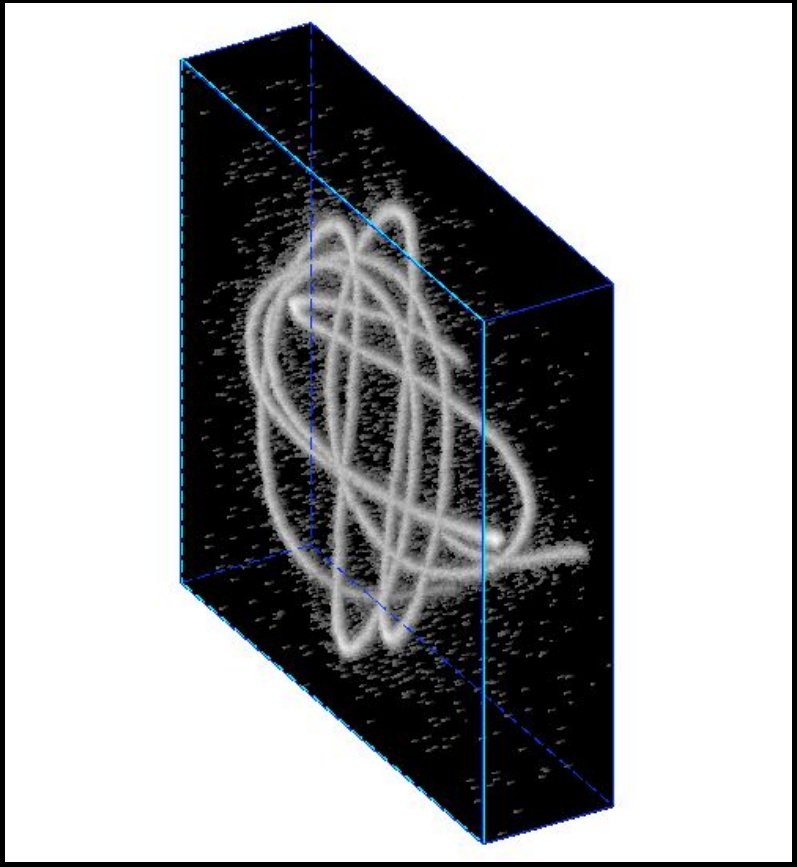
Because **Chandra dithers** during observations, the point source moves slightly across the detector over time. This motion is captured as a third dimension in the cube.

The event file is **filtered on chip coordinates** to focus on the source region.

The data is binned into a cube:

X-axis: chipx, binned by 2
Y-axis: chipy, binned by 2
Z-axis: time, divided into 50 bins

This type of cube helps visualize how the source moves across the detector over the course of the observation.



Data Manipulation Tools

- CIAO provides a powerful suite of tools for manipulating FITS and ASCII data, including four core Data Model (DM) tools: `dmlist`, `dmcopy`, `dmextract`, and `dmstat` and over 30 additional data manipulation tools

`dmlist` – Lists the contents, structure, or metadata of a file

View headers, column names, data blocks, etc.

`dmcopy` – Filters and bins data in tables or images

Create subsets of data or rebin for analysis and visualization

`dmextract` – Creates histogram-style tables from event data

Used to generate spectra (PHA files), lightcurves, or count profiles from event files

`dmstat` – Computes basic statistics on images or table columns

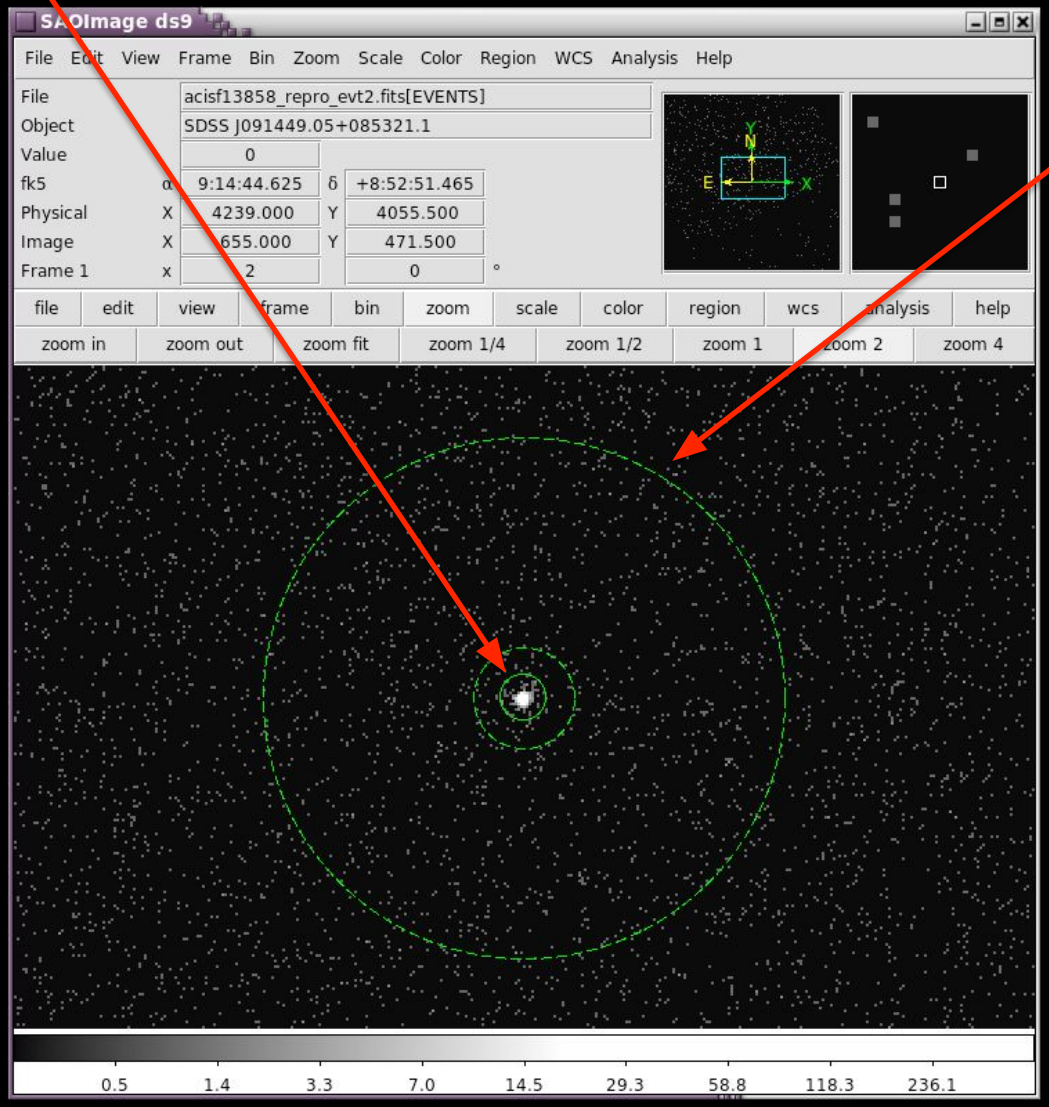
Returns mean, median, min/max, standard deviation, and pixel distributions

4. Regions

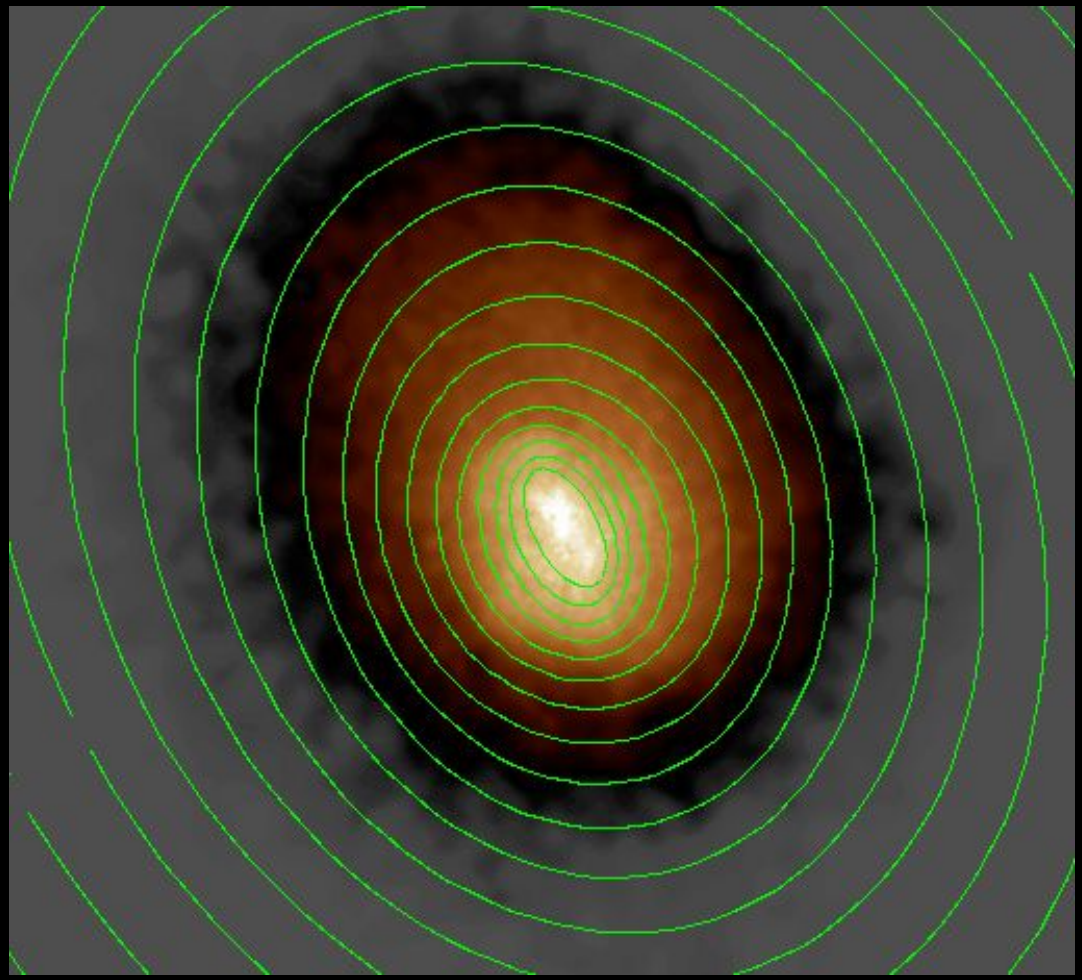
http://cxc.harvard.edu/ciao/ahelp/dmregions.html
http://cxc.harvard.edu/ciao/threads/regions/

- **Regions** are 2D spatial filters used in CIAO to include or exclude specific areas of data.
- They are commonly used to define source and background areas in images or event files.
- Regions can be saved as text files or FITS files, and are often created interactively in DS9.
- You can apply a region as a filter using the CIAO Data Model syntax,
  e.g.: [sky=region(source.reg)]
- Multiple regions can be combined using:
  - AND for intersection
  - OR for union
- This makes region filtering flexible for a wide range of spatial analysis tasks.

```
circle(9:14:49.090,+8:53:21.231,4.083")
annulus(9:14:49.074,+8:53:20.987,9.064",46.425") # background
```



```
dmellipse a1664.asm a1664.ellipses "lgrid(0.1:0.96:0.05)" step=100 clob+
```



5. Subspace

http://cxc.harvard.edu/ciao/ahelp/subspace.html

- The subspace **stores information about filters applied to a dataset** directly in the file's header.
- This metadata allows CIAO tools to automatically apply the correct calibration and maintain consistency in processing down the line.
- A file's subspace (ie. **filtering history**) can be viewed using:

```
dmlist <filename> opt=subspace
```

- This helps track how data has been filtered and ensures accurate, reproducible analysis.

Example: Viewing Subspace with dmlist

```
dmlist "acisf13736_evt2.fits[ccd_id=3,sky=circle(4324,3676,50)]" subspace
```

- Applies two filters to the event file:
  - **ccd_id=3** — selects events from CCD 3
  - **sky=circle(4324,3676,50)** — selects events within a circular region in sky coordinates
- Displays the subspace, which records these filtering operations in the file's metadata.

This allows CIAO tools to recognize what selections were applied, helping ensure correct calibration and consistent processing later on.

The command returns something like:

Data subspace for block EVENTS: Components: 1 Descriptors: 16

--- Component 1 ---

[...]
CCD_ID 3
SKY(X,Y) circle(4324,3676,50)
[...]

This shows the exact filters that were applied and stored as part of the file's "**data subspace.**"

6. Good Time Intervals

http://cxc.harvard.edu/ciao/ahelp/times.html
http://cxc.harvard.edu/ciao/dictionary/gti.html
http://cxc.harvard.edu/ciao/ahelp/dmgti.html

- Chandra event files record **timing information** in both:
  - Header keywords (e.g. TSTART, TSTOP, EXPTIME)
  - GTI blocks (Good Time Interval tables) within the file
- **GTIs define the valid time periods of an observation**—intervals during which the data are considered reliable and can be used for scientific analysis.
- The `dmgti` tool allows to create new GTI files based on custom time filters (e.g., background flares, aspect quality).
  These GTIs can then be applied to filter the data accordingly.
- GTIs are critical for ensuring that analyses are based only on **clean, usable time intervals**.

7. Python: the scripting language in CIAO

- CIAO uses **Python as its scripting language**, allowing flexible, interactive analysis without compilation.
- CIAO 4.17 includes Python 3.11 and the CIAO Conda environment fully supports Python 3.11 and associated packages
- CIAO comes with its own Python environment, but users can choose to use a custom Conda installation if needed.
- Key CIAO Python libraries:
  - **CRATES** – Provides access to the CIAO Data Model (DM) for working with FITS files
  - **TRANSFORMS** – Enables World Coordinate System (WCS) transformations and coordinate handling
- The CIAO modeling and fitting application, *Sherpa*, is available as a fully importable Python module.
- You don’t need to know Python to use Sherpa, but if you do know Python, you can script and automate complex modeling workflows with ease.

Have fun using CIAO!